



# IN2010: Forelesning 12

Repetisjon – basert på innkomne ønsker



# O-NOTASJON

# Oppg 2a 2011

```
for i = 1 to n {  
  for j = i1 to n {  
    for k = i1 to jn {  
      print (i, j, k);  
    }  
  }  
}
```

$O(n^3)$

# Opppg 1a 2015

```
int x = 0;
for (int i = 0; i < n; i++){
    for (j = 0; j < i*i; j++){
        x = x + j;
    }
}
```

*Handwritten notes:*

- A red checkmark is next to the first for loop.
- Below the inner for loop,  $n \cdot n$  is written in red.
- Below the inner for loop,  $(n \cdot n) \cdot n = O(n^3)$  is written in red.

*Handwritten notes:*

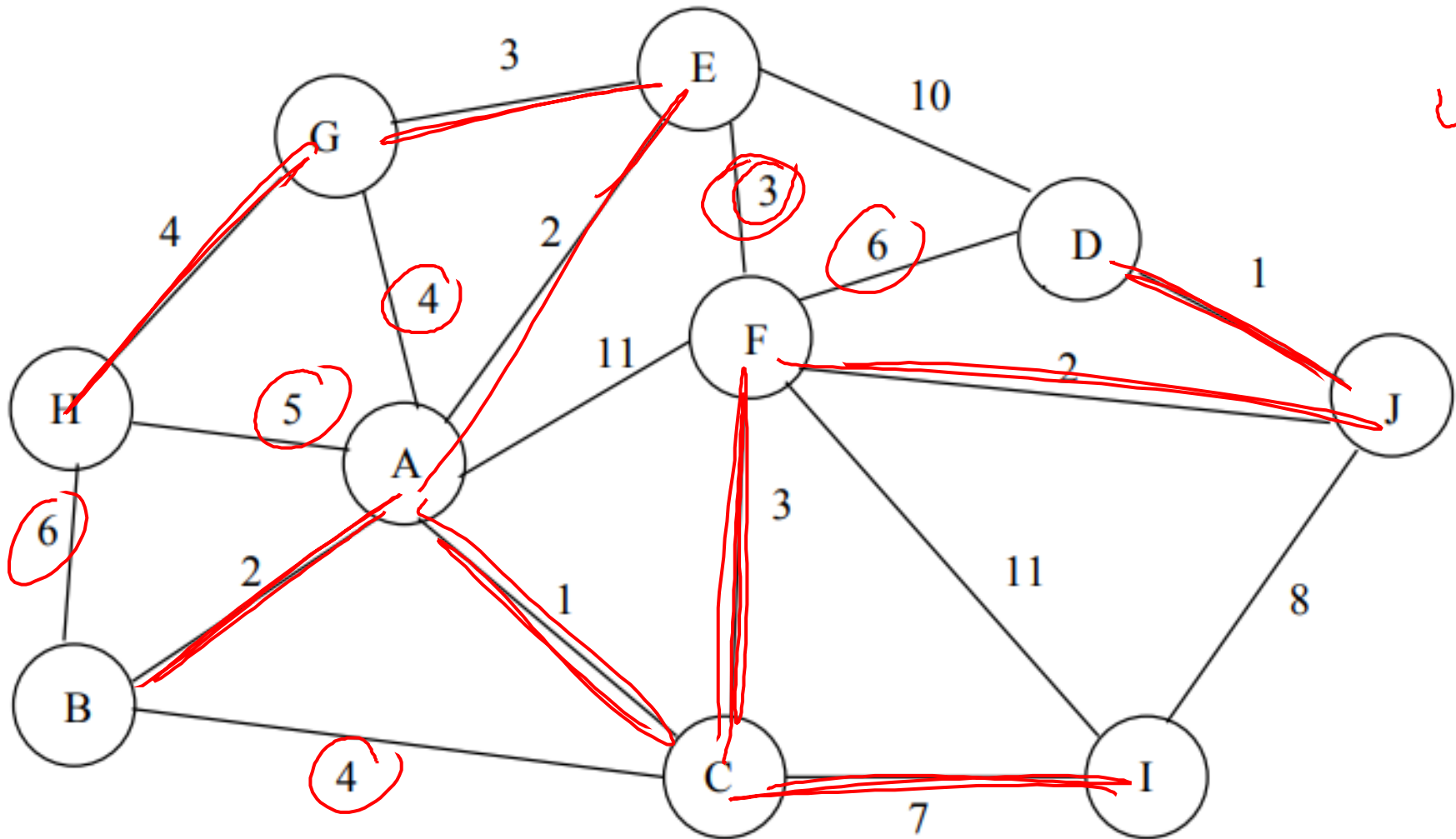
|V| |E|



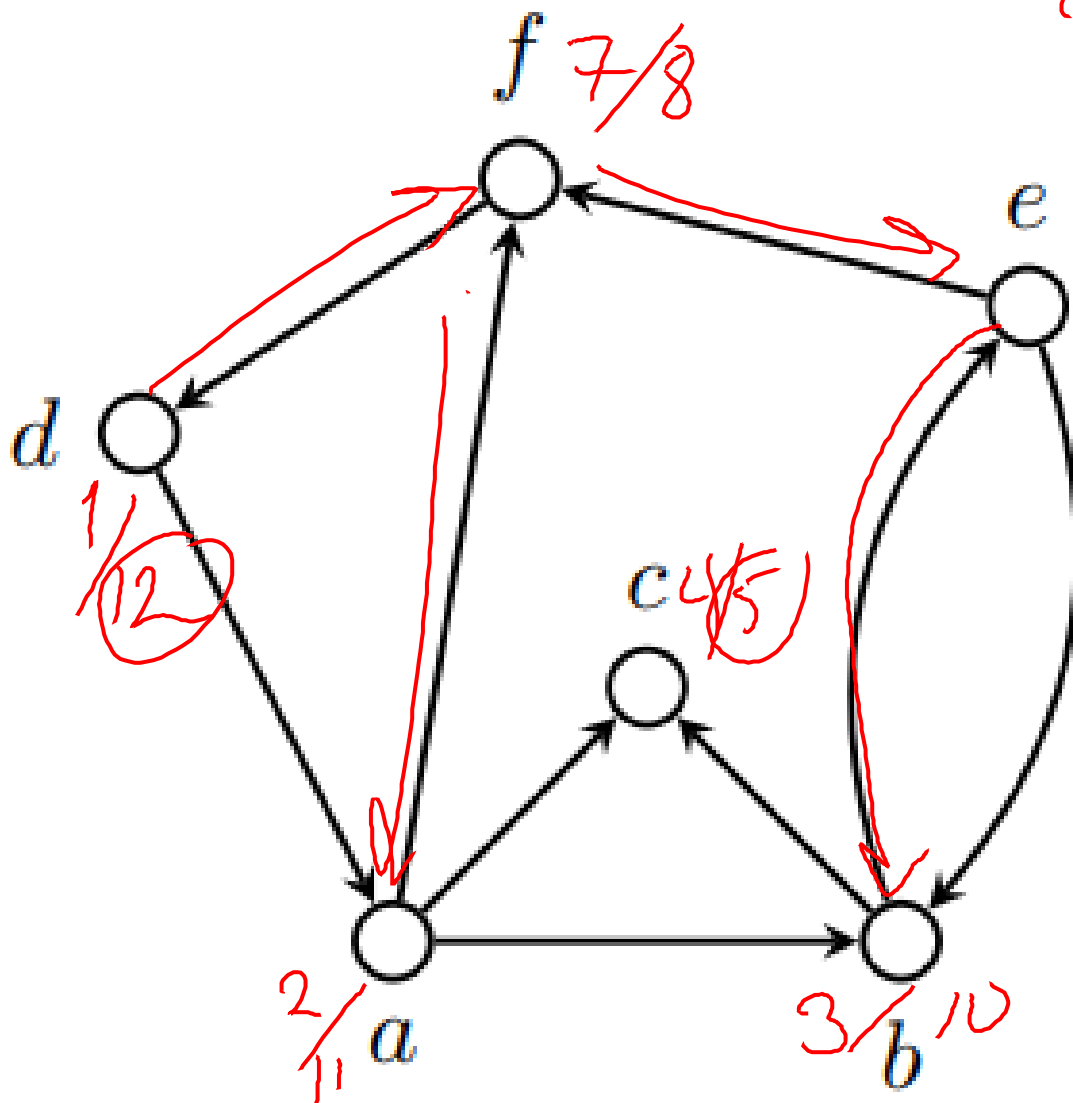
# GRAFER

r

# Oppg 5b.1 2011



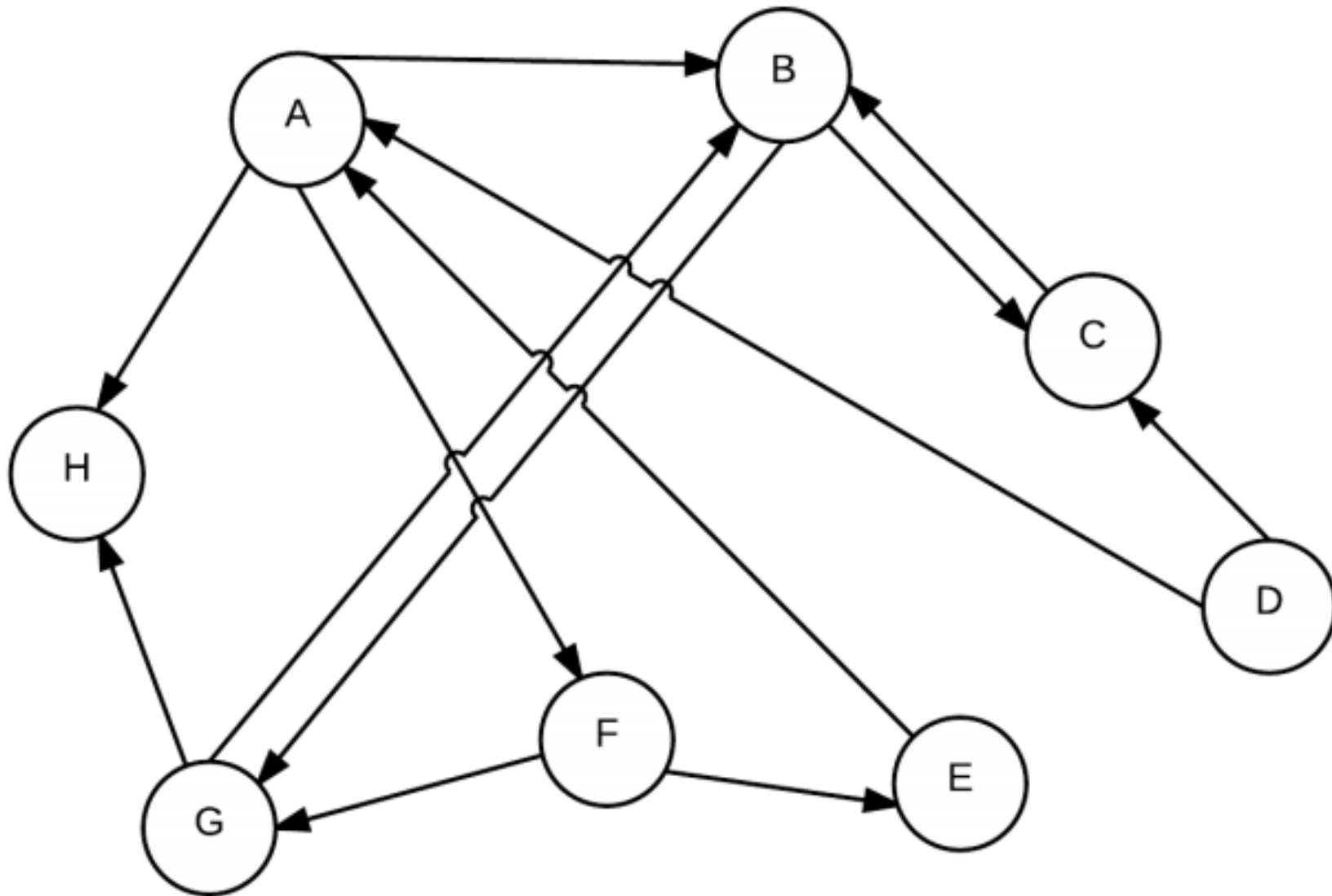
# Oppg 1b 2012



$\{d, f, a, e, b\}$

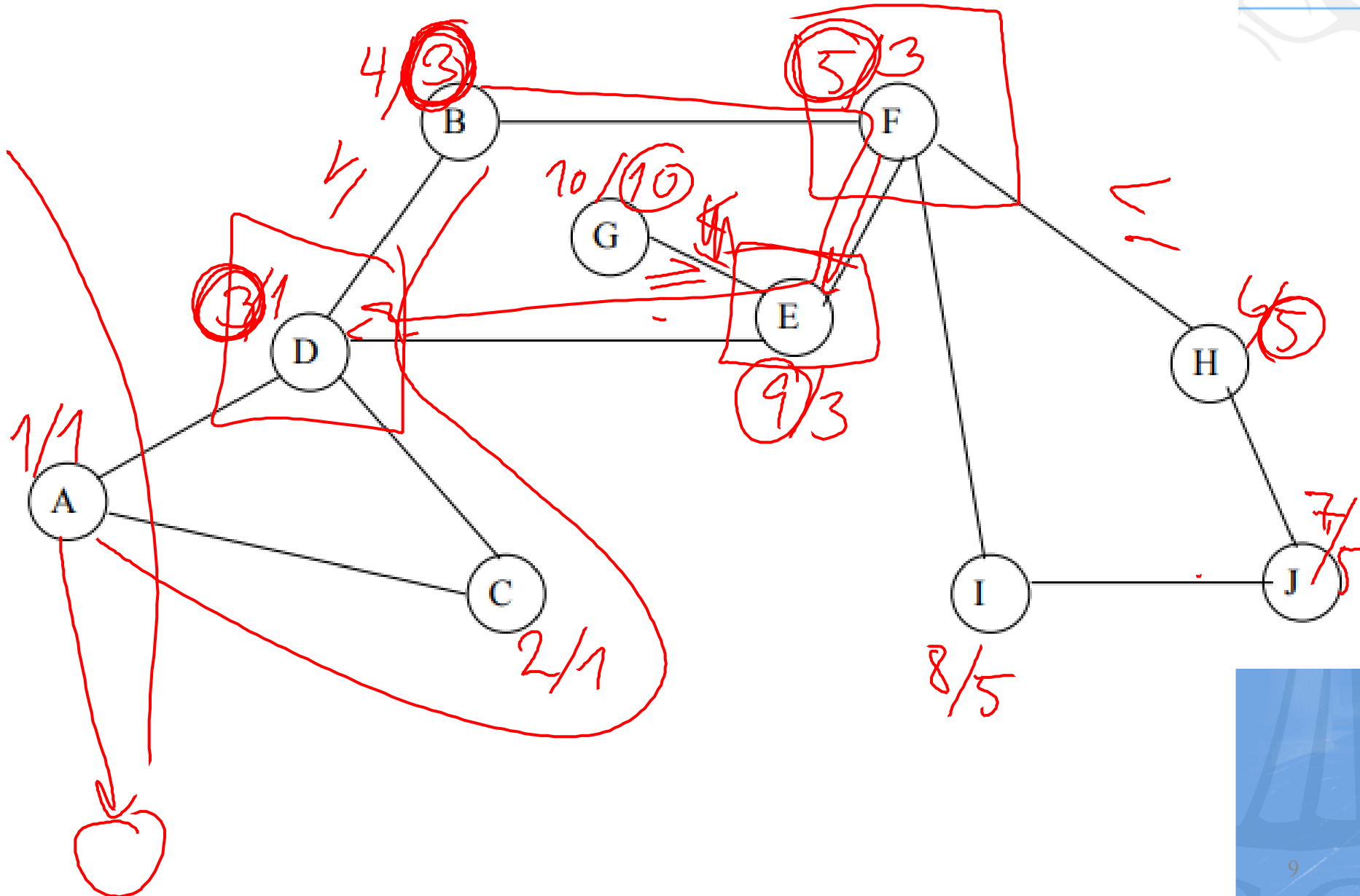
$\{c\}$

# Oppg 3a 2014





# Opppg 3a 2013





# Oppg 3b 2014

Implementer en algoritme som i tillegg til å finne korteste vei fra en gitt startnode  $s$  til alle andre noder  $w$  også beregner antall korteste veier fra  $s$  til  $w$ . [...] Du kan anta at grafen ikke har negative kanter.

# Opppg 3b 2014

Algorithm DijkstraShortestPaths( $G, v$ )

...

$D[0].min = 1$

while  $Q$  is not empty do

$u \leftarrow Q.removeMin()$

    for each vertex  $z$  adjacent to  $u$  such that  $z$  is in  $Q$  do

        if  $D[u] + w((u, z)) < D[z]$  then

$D[z] \leftarrow D[u] + w((u, z))$      $D[z].min = D[u].min$

            Change the key for vertex  $z$  in  $Q$  to  $D[z]$

        else

return the label  $D[u]$  of each vertex  $u$



# HASHING

1

# Oppg 1a 2008

Finne to svakheter ved hashfunksjonen gitt under.

```
int hashFunc(String s, int tableSize){  
  
    int hashValue=0;  
  
    for(int i = 0; i < s.length(); i++){  
        hashValue += 7 * (int) s.charAt(i);  
    }  
  
    return hashValue % tableSize;  
}
```



# Oppg 1b 2008

Forklar kort problemene som kan oppstå ved sletting av elementer når en bruker **lukket hashing**.



# Oppg 1 2009

Anta at du har en **lukket** hashtabell som er **tom**, med størrelse **11**. **Tegn hashtabellen** etter at vi har satt inn tre elementer (a,b,c) som alle hasher til samme verdi, når en bruker følgende kollisjonshåndtering.

- Lineær prøving
- Kvadratisk prøving
- Dobbel hashing\*

\* Verdien av den andre hashfunksjonen bør være spesifisert for å begrunne svaret. (Dvs. du kan spesifisere dette ved å bare gi disse en verdi for eksempel  $\text{hash2}(a) = 5$ ,  $\text{hash2}(b) = 7$  og  $\text{hash2}(c) = 7$ )



UNIVERSITETET  
I OSLO



# O-NOTASJON



# Oppg 1 2014

```
int x = 0;
for (int i = n; i >= 1; i = i/2){
    for (int j = 1; j <= i; j++){
        x = x + i + j;
    }
}
```

$O(n)$   $n + \left[ \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \dots \right] \rightarrow 2n$

```
int x = 0;
for (int j = 1; j <= n; j++){
    for (int i = n; i >= j; i = i/2){
        x = x + i + j;
    }
}
```

$O(n \log n)$   $\log n$

# Oppg 1a 2013

```
int s = 0;

for (int i = 1; i <= n; i++) {
    for (int j = 1; j <= i; j++) {
        for (int k = i; k <= j; k++) {
            s = s + k;
        }
    }
}
```

 $O(n^2)$ 

1 → 1  
2 → 1 = 0  
2 → 2  
3 → 1 = 0  
3 → 2 = 0  
3 → 3